

# **Московский государственный технический университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Базовые компоненты интернет-технологий»  
Отчет по лабораторной работе №6**

Выполнил:  
студент группы ИУ5-31Б  
Бутрим Андрей  
Александрович

Подпись: \_\_\_\_\_

Дата: \_\_\_\_\_

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Юрий  
Евгеньевич

Подпись: \_\_\_\_\_

Дата: \_\_\_\_\_

Москва, 2021 г.

# Лабораторная работа №6

## Описание задания

Разработка бота на основе конечного автомата Telegram с использованием языка Python.

Цель лабораторной работы: изучение разработки ботов в Telegram.

## Текст программы

```
import telebot
from telebot import types

TOKEN = '****'
bot = telebot.TeleBot(TOKEN)
global config
config = ['Первое число', 'Второе число', 'Действие', 'Посчитать']
global cases
cases = ['first', 'second', 'action', 'res']
global call
call = ''
global actions
actions = ['plus', 'minus']
global act
act = ['+', '-']

@bot.message_handler(commands='start')
def start(message):
    msg = 'Добро пожаловать.'
    markup = types.InlineKeyboardMarkup()
    btn = types.InlineKeyboardButton('Начать работу', callback_data='work')
    markup.add(btn)
    bot.send_message(message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == 'work')
def work(message):
    msg = 'Введите данные'
    markup = types.InlineKeyboardMarkup(row_width=1)
    for i in range(4):
        btn = types.InlineKeyboardButton(f'{config[i]}',
        callback_data=f'{cases[i]}')
        markup.add(btn)
    btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
    markup.add(btn)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == cases[0] or
message.data == cases[1])
def nums(message):
    global call
    call = cases[cases.index(message.data)]
    msg = 'Введите число'
    bot.send_message(message.message.chat.id, msg)

@bot.callback_query_handler(lambda message: message.data == cases[2])
def action(message):
```

```

msg = 'Выберите действие'
markup = types.InlineKeyboardMarkup(row_width=2)
btn = types.InlineKeyboardButton('+', callback_data='plus')
btn1 = types.InlineKeyboardButton('-', callback_data='minus')
markup.add(btn, btn1)
bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data in actions)
def act(message):
    if message.data == actions[0]:
        config[2] = actions[0]
    else:
        config[2] = actions[1]
    markup = types.InlineKeyboardMarkup(row_width=1)
    msg = 'Введите данные'
    for i in range(4):
        if not config[i].isdigit() and not config[i] in actions:
            btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
            markup.add(btn)
    btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
    markup.add(btn)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == cases[3])
def res(message):
    if config[0].isdigit() and config[1].isdigit() and (config[2] in
actions):
        if config[2] == 'plus':
            config[3] = int(config[0]) + int(config[1])
            msg = f'{int(config[0])} + {int(config[1])} = {config[3]}'
        else:
            config[3] = int(config[0]) - int(config[1])
            msg = f'{int(config[0])} - {int(config[1])} = {config[3]}'
        markup = types.InlineKeyboardMarkup()
        btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
        markup.add(btn)
        bot.send_message(message.message.chat.id, msg, reply_markup=markup)
    else:
        msg = 'Недостаточно данных'
        markup = types.InlineKeyboardMarkup()
        for i in range(4):
            if not config[i].isdigit() and not config[i] in actions:
                btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
                markup.add(btn)
        btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
        markup.add(btn)
        bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == 'reset')
def reset(message):
    msg = 'Данные сброшены.'
    global config
    config = ['Первое число', 'Второе число', 'Действие', 'Посчитать']
    markup = types.InlineKeyboardMarkup()
    btn = types.InlineKeyboardButton('Продолжить', callback_data='work')
    markup.add(btn)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

```

```

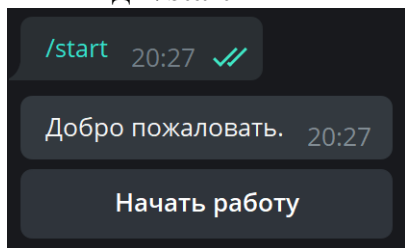
@bot.message_handler(content_types='text')
def text(message):
    if (call == cases[0] or call == cases[1]) and message.text.isdigit():
        if call == cases[0]:
            config[0] = message.text
            bot.send_message(message.chat.id, f'Вы ввели первое число
{int(message.text)}')
        elif call == cases[1]:
            config[1] = message.text
            bot.send_message(message.chat.id, f'Вы ввели второе число
{int(message.text)}')
        markup = types.InlineKeyboardMarkup(row_width=1)
        msg = 'Введите данные'
        for i in range(4):
            if not config[i].isdigit() and not config[i] in actions:
                btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
                markup.add(btn)
            btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
            markup.add(btn)
        bot.send_message(message.chat.id, msg, reply_markup=markup)
    if (call == cases[0] or call == cases[1]) and not message.text.isdigit():
        msg = 'Ошибка, введите число'
        bot.send_message(message.chat.id, msg)

bot.polling(none_stop=True)

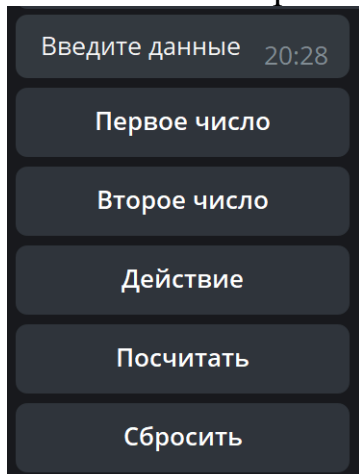
```

## Примеры выполнения программы

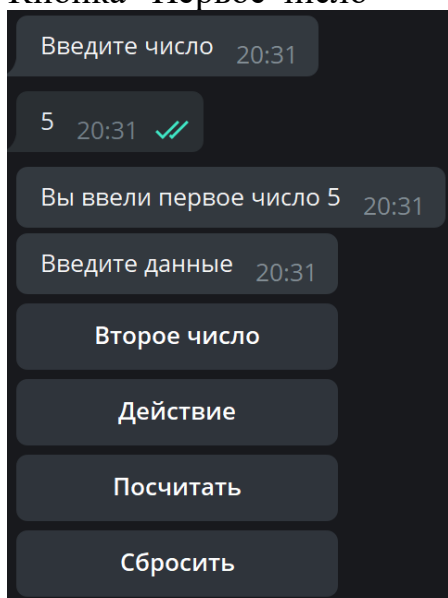
Команда /start



Кнопка “Начать работу”



Кнопка “Первое число”



### Кнопка “Второе число”

Введите число 20:31

3 20:31 ✓✓

Вы ввели второе число 3 20:31

Введите данные 20:31

Действие

Посчитать

Сбросить

### Кнопка “Действие”

Выберите действие 20:32

+ -

Введите данные 20:32

Посчитать

Сбросить

### Кнопка “Посчитать”

5 + 3 = 8 20:32

Сбросить

### Кнопка “Сбросить”

Данные сброшены. 20:33

Продолжить

Введите данные 20:33

Первое число

Второе число

Действие

Посчитать

Сбросить